A New Class of Traceability Schemes for Protecting Digital Content against Illegal Re-distribution

Xin-Wen Wu^, and Abdul Sattar^*

^Institute for Integrated and Intelligent System (IIIS)
^School of Information and Communication Technology, Griffith University, Gold Coast, QLD 4222, Australia
*National ICT Australia

Summary
Traceability schemes are used to protect the rights of intellectual-property owners against illegal re-distribution. In order to trace a pirate copy back to a user who has been involved in making and re-distributing the pirate copy, a traceability scheme requires a code which contains a sufficient number of codewords and a traitor-tracing algorithm. Codes with the identifiable parent property (IPP codes) and traceability codes (TA codes) have been extensively studied. IPP codes include all TA codes as special instances. However, TA codes usually implement efficient tracing algorithms, while IPP codes do not always have efficient tracing algorithms. A major theoretical challenge is to derive more codes which have efficient tracing algorithms. The contributions of this paper include a new class of traceability codes, as well as an efficient tracing algorithm for the new traceability codes. The proposed tracing algorithm has the same complexity as the traditional tracing algorithm and can outperform the traditional tracing algorithm.

Key words:
Digital content protection, Traceability scheme, Traceability code, Code with identifiable parent property, generalized traceability code, tracing algorithm

1. Introduction

With the increasing popularity of digital products (e.g., digital documents, images, music, movies, and software), there is a strong desire to develop effective solutions to the problem of protecting digital products against illegal re-distribution [2], [5], [16]. Some approaches to detecting copyright infringements include monitoring P2P networks and blocking the data transfer and/or identifying the end users [12]. However, since some data transferred using P2P networks is licensed to permit this, a heavy-handed approach to blocking all traffic is not appropriate. Also, while it is possible to monitor the content of (unencrypted) P2P traffic to search for matches on particular hashes of known copyrighted data, maintaining and distributing a list of all such files to all routers in real-time is not feasible. Thus, it makes more sense to encapsulate intellectual property rights within the digital product, and to ensure that access rights can be managed.

Traditional encryption schemes alone do not provide an effective solution to this problem, because they do not prevent authorized users from transferring the clear-text content to unauthorized users. Also, once the transfer has been completed, there is no means to trace the unauthorized use of the content back to the source of the leak by any encryption scheme [2], [7]. Traceability schemes are used to protect the rights of intellectual-property owners against illegal re-distribution. With a traceability scheme the owner of a digital product can trace a pirate copy back to users who have been involved in making and re-distributing the pirate copy (we call such a user a traitor).

An effective traceability scheme consists of a well-designed code (which is a collection of codewords) and a traitor-tracing algorithm. In a model for collusion-resistant traitor tracing proposed in [8], a unique codeword is inserted (by using an invisible watermarking procedure, for example) into each copy of the digital product before being sold or distributed. Then, by the traitor-tracing algorithm, with the codeword retrieved from the observed pirate copy (the codeword may have been altered by the users who colluded for making the pirate copy), at least one of the traitors will be identified. Two types of codes have been extensively studied [1], [2], [4], [5], [6], [7], [8], [10], [15], [16], [17]. They are codes with the identifiable parent property (IPP codes) and traceability codes (TA codes). While by definition of IPP codes, at least one traitor is theoretically identifiable, it is not always guaranteed that an efficient tracing algorithm is available. On the other hand, many TA codes can implement efficient tracing algorithms.

The family of IPP codes includes all TA codes as special members; and the currently known TA codes with efficient tracing algorithms form a small subset of the family of IPP codes. As shown in the literature, there are many IPP codes which are not TA codes [16], [17]. For those IPP codes, no efficient tracing algorithms are available. Therefore, a major theoretical challenge is to derive more codes which have efficient tracing algorithms. In this paper, generalizing traditional TA codes, we derive

Manuscript received December 5, 2011
Manuscript revised December 20, 2011

a new class of codes. While the generalized TA codes still retain the identifiable parent property, they extend traditional TA codes and contain new codes. Moreover, adapting a decoding algorithm from our previous work [18],[19], we derive an efficient tracing algorithm for the new traceability codes. We will show that the tracing algorithm can outperform the traditional tracing algorithm [16].

Traditional TA codes are defined with respect to the Hamming metric (see [16], [17], for example). We generalize this definition by defining traceability codes with respect to any well-defined metric. We will prove that the generalized traceability codes with respect to any metric still have the identifiable parent property (that is, they belong to IPP codes). As there are many well-defined metrics different from the Hamming metric (for instances, edit distance, Euclidean distance, and Lee distance [3], [11], [14], [18], [19]), the family of generalized TA codes is obviously larger than that of traditional TA codes and contains new codes. To be more explicit, we then focus on TA codes with respect to the Lee metric. By giving a precise example, we will show that there exist Lee-metric TA codes which are new codes, that is, they are not traditional TA codes. We will then present an efficient tracing algorithm for the new Lee-metric TA codes.

The rest of the paper is organized as follows. In Section 2, we review the background. In Section 3, we generalize TA codes and present new TA codes. In Section 4, we present an efficient tracing algorithm for the new Lee-metric TA codes. We analyze the performance and complexity of the proposed tracing algorithm in Section 5. Concluding remarks will be given in Section 6.

2. Background

Let A be an alphabet with $|A|=q.$ A code C of length n over A is a subset of A^n, where A^n is the set of all n-tuples with components in $A.$ If $|C|=M$ (that is, C has M elements), we call C a (n,M) code. An element $\mathbf{c}$ of C, i.e., $\mathbf{c}\in C$, is called a codeword. In practical applications, each codeword uniquely corresponds to an authorized user of a digital product (or a legal copy of the digital product). A subset of the code, $D\subseteq C$, corresponds to a group of users. A pirate copy (or simply called pirate) corresponds to a n-tuple in A^n. If a group of users, D, have colluded to produce a pirate copy, D is called a coalition. An element of a coalition is called a traitor.

For a coalition $D,$ a n-tuple $\mathbf{x}=(x_1,x_2,\ldots,x_n)$ is called a descendant of $D,$ provided that for all x_i (i = 1, . . . , n), $x_i=a_i,$ for some $(a_1,\ldots,a_i,\ldots,a_n)\in D.$

The set of all descendants of D is denoted as $desc(D).$ The size of a coalition, $|D|$, is an important parameter, which can affect the effectiveness of a traceability scheme. Letting t be any positive integer, we define

$$desc_t(C)=\bigcup_{D\subseteq C\text{, and }|D|\le t} desc(D).$$

That is, $desc_t(C)$ is the set of n-tuples that can be produced by a coalition of size at most t.

In the literature the Hamming metric has been used to define traceability codes. For any $\mathbf{x}=(x_1,x_2,\ldots,x_n)$ and $\mathbf{y}=(y_1,y_2,\ldots,y_n)\in A^n$, the Hamming distance between $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$d_H(\mathbf{x},\mathbf{y})=|\{i\mid x_i\neq y_i\}|$$

that is, the number of coordinates where $\mathbf{x}$ and $\mathbf{y}$ differ.

Suppose C has s subsets of size at most t. It is clear that when $|C|=M$, $s=\sum_{i=1}^{t}\binom{M}{i}$. We are now ready to define codes with the identifiable parent property (IPP codes) and traceability codes (TA codes).

Definition 1. Suppose C is a code of length *n*. Let $t\ge 2$ be an integer. Let $D_i\subseteq C,\ i=1,\ldots,s,$ be all the subsets of C with $|D_t|\le t.$

(1) C is a *t*-IPP code, provided that for all $\mathbf{x}\in desc_t(C)$ the following is true

$$\bigcap_{\{i|\mathbf{x}\in desc(D_i),|D_i|\le t\}} D_i\neq\phi.$$

(2) C is a *t*-TA code, provided that for all D_i and any $\mathbf{x}\in desc(D_i)$ there exists at least one codeword $\mathbf{y}\in D_i$ such that

$$d_H(\mathbf{x},\mathbf{y})<d_H(\mathbf{x},\mathbf{z}),\quad\text{for any }\mathbf{z}\in C-D_i.$$

Remark 1. Consider a *t*-IPP code. If a pirate copy $\mathbf{x}$ was produced by some coalition D of size at most *t*, then the traitors must be in the intersection $\bigcap_{\{i|\mathbf{x}\in desc(D_i),|D_i|\le t\}} D_i$. By definition, this intersection is not empty. Thus, the traitors are identifiable. However, the definition does not provide any information on a procedure for constructing the above-mentioned intersection or searching the elements of the intersection. Suppose C is a (n,M) code. By the exhaustive search, it requires $O(\binom{M}{t})$ comparisons to find a traitor. If the code of length n is defined over an alphabet A of size q, then $M = O(q^n)$; and thus the exhaustive traitor search has an exponential time-complexity in code length n.

On the other hand, t-TA codes can have a more efficient traitor tracing procedure. By the maximum likelihood principle, the codewords which are closest to the pirate are traitors. By definition of t-TA codes, at least one traitor can be found by comparing the Hamming distance between the pirate copy and the codewords. This procedure will require M comparisons, and is much more efficient than the exhaustive search, which requires $O(\binom{M}{t})$ comparisons. But as $M = O(q^n)$, this procedure still has an exponential complexity in code length n. Error-correcting codes have been used to attain TA codes [1], [2], [4], [5], [10], [15], [16], [17]. A lot of block error-correcting codes, including Reed-Solomon and algebraic-geometric codes, are TA codes, when these error-correcting codes have large minimum distances and small code rates [4], [5], [16], [17]. In [16], the well-known list decoding of Reed-Solomon and algebraic-geometric codes has been adapted to an efficient tracing algorithm. Suppose the TA codes are of length n, the tracing algorithm in [16] has a time-complexity which is polynomial in code length n. This has been the first polynomial-complexity tracing algorithm.

The following is an important result which shows that all t-TA codes satisfy the identifiable parent property, that is, the family of t-IPP codes includes all t-TA codes as special members. The proof of this result can be found in [17].

Proposition 1. Any t-TA code is a t-IPP code.

On the other hand, there are many t-IPP codes that are not t-TA codes (see [16], [17]). In the following sections, we will generalize TA codes to attain new traceability codes; we will also propose a polynomial-complexity tracing algorithm, which outperforms the tracing algorithm in [16].

3. Generalized Traceability Codes

Given a set S, a function d from $S \times S$ to $\Re$ (the set of real numbers) is a distance (or called metric) if and only if it satisfies the following three properties:

(1) For any x, y ∈ S, d(x, y) ≥ 0, and d(x, y) = 0 if and only if x = y.
For any x, y ∈ S, d(x, y) = d(y, x).
For any x, y, z ∈ S, d(x, y) ≤ d(x, z) + d(z, y).

It is easy to verify that the Hamming distance is a well-defined distance. There are many well-defined distances (for examples, edit distance, Euclidean distance, and Lee distance [3], [11], [14], [18], [19]) which are different from the Hamming distance.

We are now ready to define generalized t-TA codes with respect to any well-defined distance.

Definition 2. (Generalized Traceability Codes) Suppose C is a code of length n. Let t ≥ 2 be an integer. Let $D_i \subseteq C,\ i = 1,\ldots,s,$ be all the subsets of C of size at most t, that is, $|D_t| \leq t.$ We call C a generalized t-TA code (or t-GTA code for short), provided that there exists a well-defined distance d, such that for all D_i and any $\mathbf{x} \in desc(D_t),$ there exists at least one codeword $\mathbf{y} \in D_t$ such that

$$d(\mathbf{x},\mathbf{y}) < d(\mathbf{x},\mathbf{z}), \quad \text{for any} \quad \mathbf{z} \in C - D_i.$$

Comparing with the definition of t-TA codes (i.e., Definition 1), this obviously generalizes t-TA codes as the Hamming distance is a special instance of distances. The generalized TA codes will bring us more traceability codes as well as new efficient tracing algorithms based on other metrics as we will see in the following sections.

We now prove that the generalized traceability codes still have the identifiable parent property, that is, for the generalized traceability codes the traitors are identifiable.

Theorem 1. Any generalized *t*-TA code is a *t*-IPP code.

Proof: Suppose C is a generalized t-TA code with respect to a well-defined distance d. Let $\mathbf{x} \in desc_t(C).$ Then there is a coalition $D_i \subseteq C,$ with $|D_i| = t$ and $\mathbf{x} \in desc(D_i).$ Let $\mathbf{y} \in D_i$ such that $d(\mathbf{x},\mathbf{y}) \leq d(\mathbf{x},\mathbf{z})$ for every $\mathbf{z} \in D_i.$ Then $d(\mathbf{x},\mathbf{y}) \leq d(\mathbf{x},\mathbf{z})$ for any z ∈ C by the definition of generalized t-TA code.

Next, we prove that for any $D_j \subseteq C$ with $|D_j| \leq t,$ if $\mathbf{x} \in desc(D_j)$ then $\mathbf{y} \in D_j.$ In fact, if $\mathbf{y} \notin D_j$, then there is a $\mathbf{w} \in D_j$ such that $d(\mathbf{x},\mathbf{w}) < d(\mathbf{x},\mathbf{y})$ by definition of generalized t-TA codes. This contradicts the fact that $d(\mathbf{x},\mathbf{y}) \le d(\mathbf{x},\mathbf{z})$ for any z ∈ C. As $\mathbf{y} \in D_j$ for any $D_j \subseteq C$ with $| D_j | \le t$ and $\mathbf{x} \in desc(D_j)$, we conclude that the intersection $\bigcap_{\{ j|\mathbf{x}\in desc(D_j),|D_j|\le t\}} D_j$ is not empty. Therefore, C is a t-IPP code.

In the following, by giving an example, we will show that the family of generalized traceability codes contains new codes which are not traditional traceability codes.

In the rest of the paper, we focus on the Lee metric. Let Zp be the ring of integers modulo p, where p is a prime. From basic algebra we know Zp is actually a field, which is also denoted as F_p. For any $a \in F_p$, the Lee value of a, denoted by $|a|$, is the nonnegative integer $\min\{a, p-a\}$. For a n-tuple $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in F_q^n$, the Lee weight is defined as $\|\mathbf{x}\|_L = \sum_{i=1}^{n} |x_i|$. The Lee distance between two n-tuples $\mathbf{x}$ and $\mathbf{y}$ in F_q^n, denoted by $d_L(\mathbf{x},\mathbf{y})$, is defined as the Lee weight of $\mathbf{x}-\mathbf{y}$. It is easy to verify that the Lee distance is a well-defined distance.

The Lee distance defined above can be extended to any finite field (see [18], [19]). When we consider the binary field F_2 and ternary field F_3, the Lee distance coincides with the Hamming distance. However, for other finite fields, they are different.

The following is an example of new traceability codes, which is a generalized *t*-TA code with respect to the Lee distance. But it is not a traditional *t*-TA code.

Example 1. Consider F_{11} the finite field of 11 elements. Let C be the following code of length 3 over the finite field F_{11}.

$$C = \{(1,0,0),(4,1,1),(5,1,1)\}.$$

The code C is a 2-IPP code; while it is not a 2-TA code.

In fact, the symbols in the first position of the codewords are distinct. Thus, for any pirate copy $\mathbf{x} \in F_{11}^3$, every coalition of size at most 2 which can produce $\mathbf{x}$ must contain the codeword which has the same first coordinate with the pirate copy as a common codeword. Thus, C is a 2-IPP code.

Consider a pirate copy $\mathbf{x} = (1,1,1)$. Obviously, it is a descendant of the following coalition

$$D = \{(1,0,0),(4,1,1)\}.$$

Now,

$$d_H((1,1,1),(1,0,0)) = 2 \text{ and}$$

$$d_H((1,1,1),(4,1,1)) = 1.$$

$C - D = \{(5,1,1)\}$ and $d_H((1,1,1),(5,1,1)) = 1.$

Thus, there is no codeword $\mathbf{y} \in D$ satisfying

$$d_H(\mathbf{x},\mathbf{y}) < d_H(\mathbf{x},\mathbf{z}), \quad \text{for any } \mathbf{z} \in C - D.$$

Therefore, C is not a 2-TA code.

Now, let us consider the Lee distance d_L. The following are the subsets of C of size 2:

$D_1 = \{(1, 0, 0), (4, 1, 1)\}$,
$D_2 = \{(1, 0, 0), (5, 1, 1)\}$, and
$D_3 = \{(4, 1, 1), (5, 1, 1)\}$.

We will show that C is a generalized 2-TA code with respect to the Lee distance, by proving that all D_i satisfy the condition: For any $\mathbf{x}$ ∈ *desc*(D_i), there is a codeword $\mathbf{y}$ ∈ D_i such that $d_L(\mathbf{x}, \mathbf{y}) < d_L(\mathbf{x}, \mathbf{z})$, for any $\mathbf{z} \in C - D_i$.

First, considering D_1 we have

desc(D_1) = {(1, 0, 0), (4, 1, 1), (1, 0, 1), 1, 1, 0), (1, 1, 1), (4, 0, 0), (4, 0, 1), (4, 1, 0)}.

For (1, 0, 0) and (4, 1, 1), as they are in D_1, they have Lee distance 0 to themselves. Thus, the condition above is satisfied. Look at (1, 0, 1),

$$d_L((1,0,1),(1,0,0)) = 1 < d_L((1,0,1),(4,1,1) = 4,$$

and

$$d_L((1,0,1),(1,0,0)) = 1 < d_L((1,0,1),(5,1,1) = 5.$$

Therefore, for $\mathbf{x}$ = (1, 0, 1) ∈ *desc*(D_1), the codeword (1, 0, 0) ∈ D_1 is such a $\mathbf{y}$ satisfying the above condition. Now, for any of (1, 1, 0), (1, 1, 1), (4, 0, 0), (4, 0, 1), (4, 1, 0), we can similarly find a $\mathbf{y}$ ∈ D_1 such that $d_L(\mathbf{x}, \mathbf{y}) < d_L(\mathbf{x}, \mathbf{z})$, for any $\mathbf{z} \in C - D_1$. Therefore, D_1 satisfies the above condition. Similarly, we can prove that D_2 and D_3 both satisfy the above condition.

Therefore, C is a generalized 2-TA code with respect to the Lee distance.

4. Efficient Lee-Metric Tracing Algorithm

Reed-Solomon (RS) and algebraic-geometric (AG) codes have been used as traceability codes; and the list decoding algorithm with respect to the Hamming metric has been adapted to an efficient tracing algorithm [16]. In this section we address RS-traceability codes and AG-traceability codes with respect to the Lee metric. We will drive an efficient tracing algorithm based on a Lee-metric decoding algorithm in [18], [19].

To be self-contained, let us have a brief review of RS and AG codes. As AG codes are a generalization of RS codes, to simplify the statement, the Lee-metric tracing algorithm will be presented for RS-traceability codes. (The algorithm can be straightforwardly adapted to AG-traceability codes).

A RS code, $C_{(n,k)}$, over F_p, the finite filed with p elements, is defined as

$$C_{(n,k)} = \{(f(\alpha_1),\dots, f(\alpha_n)) \mid f(X) \in F_p[X], \deg(f(X)) < k\}$$

where $\alpha_1,\dots,\alpha_n$ are n distinct nonzero elements in F_p, and $F_p[X]$ stands for the set of polynomials in X with coefficients in F_p. This code is of length n and dimension k (see [16], for example).

Let X be a nonsingular, absolutely irreducible curve of genus g in the m-dimensional projective space $\mathbf{P}^m$ over the finite field F_p. Suppose $\{P_1, P_2,\dots, P_n\}$ is a set of rational points of X. Let $D = P_1 + P_2 + \dots + P_n$, and let G be a divisor of X satisfying $\sup(D) \cap \sup(G) = \phi$. An AG code, $C(D,G)$, over F_p, is defined as

$$C(D,G) = \{(f(P_1), f(P_2),\dots, f(P_n)) \mid f \in L(G)\}.$$

If $\rho = \deg(G) < n$, then $C(D,G)$ has length = n and dimension $\geq \rho - g + 1$. If moreover $2g - 2 < \rho < n$, then the dimension of this code is exactly $\rho - g + 1$ (see [16], for example).

Let $\mathbf{x} = (x_1,\dots,x_n)$ be retrieved from an observed pirate copy (for simplicity we call $\mathbf{x}$ a pirate). A tracing algorithm is to find out some or all of the codewords within certain distance, say τ, from the pirate. As we have seen from definition of RS codes, each codeword uniquely corresponds to a univariate polynomial in $F_p[X]$ (which we call a codeword polynomial). The tracing algorithm will find out all codeword polynomials that correspond to the codewords within distance τ from the pirate, through two crucial steps: Interpolation Step, that is, a step of constructing a bivariate polynomial $Q(X,Y)$ (by an interpolation procedure) that contains the codeword polynomials as its factors, and Factorization Step, that is, a step of factorizing $Q(X,Y)$ to find all the codeword polynomials that correspond to the codewords within distance τ from $\mathbf{x}$.

As a preprocessing step of the Lee-metric tracing algorithm, we need a number of points on the plane over F_p, which are constructed as follows. Let u be an integer with $0 \leq u \leq (p-1)/2-1$. For each $i \in \{1, 2, \dots, n\}$, we define a set S_i of $2u+1$ points as

$$S_i = \{(\alpha_i, x_i - u), (\alpha_i, x_i - u + 1), \dots, (\alpha_i, x_i), (\alpha_i, x_i + 1), \dots, (\alpha_i, x_i + u)\}$$

where α_i is the i-th point used to define the RS code, and x_i is the i-th coordinate of the pirate $\mathbf{x}$. Denote

$$\mathbf{S} = S_1 \cup S_2 \cup \dots \cup S_n.$$

Obviously $\mathbf{S}$ has $n(2u+1)$ points. Denote the points of $\mathbf{S}$ by $(\beta_1, z_1), (\beta_2, z_2), \dots, (\beta_N, z_N)$, where $N = n(2u+1)$. We are now ready to present the tracing algorithm.

Algorithm 1 (Lee-metric Tracing Algorithm)

<u>*Inputs*</u>: The pirate $\mathbf{x}$, and the set of points $\mathbf{S}$, and the following parameters:

$$n,\ k,\ u,\ \tau, \text{ and } m = n - \left\lceil \frac{\tau}{u+1} \right\rceil.$$

Step 1: Choose two integers $\gamma,\ l$ such that $\gamma m > 1$ and $(2u+1)n\binom{\gamma+1}{2} < \frac{l(l+2)}{2(k-1)}$.

Step 2: Find a nonzero bivariate polynomial $Q(X,Y) = \sum a_{ij} X^i Y^j$ with coefficients $a_{ij} \in F_p$ such that $\max\{i + jk \mid a_{ij} \neq 0\} \leq l$, through the following interpolation procedure:

Compute $Q(X,Y)$ satisfying that for $i = 1,2,\dots,N$, (β_i, z_i) is a zero of $Q(X,Y)$ of multiplicity γ.

Step 3: Find all unvariate polynomials $f(X) \in F_p[X]$ of $\deg(f(X)) < k$, such that $Y - f(X)$ is a factor of $Q(X,Y)$, through factorizing $Q(X,Y)$ or an efficient root-finding algorithm (see [20]). Then for each codeword $\mathbf{c} = (f(\alpha_1), ..., f(\alpha_n)$ corresponding to $f(X)$, check if $d_L(\mathbf{x}, \mathbf{c}) \le \tau$. If so, include $\mathbf{c}$ in the output.

The correctness of the algorithm has been proved in our previous work [18] as a Lee-metric decoding algorithm for error-correcting codes. The purpose of constructing the set $\mathbf{S}$ (in the preprocessing step) and choosing the integers γ and l (in the first step) is to ensure the algorithm working well and to optimize the performance of the algorithm. It is clear that if the algorithm works for a larger τ, then it can find more traitors. Thus, we will measure the traitor-tracing performance by the maximum τ for which the algorithm works. As a corollary of Theorem 2 of [18], we have the following results.

Theorem 2. Suppose $\mathbf{x}$ is a pirate. Algorithm 1 (or an adapted algorithm for AG-traceability codes) finds all traitors $\mathbf{c} \in C$ that satisfy

$$d_L(\mathbf{x}, \mathbf{c}) \le \tau,$$

where τ is

$$\tau = (u+1)(n - \left\lfloor \sqrt{(2u+1)n(k-1)} \right\rfloor - 1) \quad (4.1)$$

for any Lee-metric RS-traceability code $C_{(n,k)}$ of length n and dimension k, and

$$\tau = (u+1)(n - \left\lfloor \sqrt{(2u+1)n(k+g-1)} \right\rfloor - 1) \quad (4.2)$$

for any Lee-metric AG-traceability code $C(D, G)$ defined on a curve of genus g and of length n and dimension $k = \rho - g + 1$, where the integer u satisfies $0 \le u \le (p-1)/2 - 1$.

In next section, we will analyze the performance and the complexity of the algorithm, and compare it with the traditional tracing algorithm in [16].

5. Performance and Complexity

We first give an example to show that the Lee-metric tracing algorithm can find more traitors than the traditional Hamming-metric tracing algorithm in [16]; we than give a general analysis of the performance and complexity of the Lee-metric tracing algorithm in the second subsection.

5.1 An Example

Let us consider a RS-traceability code $C = C_{(n,k)}$ over the alphabet F_{13}. Let $n = 12$ and $k = 2$. Then the code C is a 3-TA code. For any $\mathbf{x} \in desc_3(C)$, the Hamming-metric tracing algorithm can find all the traitors $\mathbf{c} \in C$ satisfying the following (see [16])

$$d_H(\mathbf{x}, \mathbf{c}) \le n - \left\lfloor \sqrt{n(k-1)} \right\rfloor - 1 = 8.$$

Now consider a coalition $D = \{\mathbf{c}_1, \mathbf{c}_2\}$ of 2 users where

$$\mathbf{c}_1 = (0,0,0,0,0,0,0,0,0,0,0,0), \text{ and}$$

$$\mathbf{c}_2 = (1,1,1,1,1,1,1,1,1,1,1,1).$$

$\mathbf{x} = (1,1,1,1,1,1,1,1,1,1,0,0)$ is a pirate produced by D, i.e., $\mathbf{x} \in desc(D)$. We have

$$d_H(\mathbf{x}, \mathbf{c}_1) = 10, \quad \text{and} \quad d_H(\mathbf{x}, \mathbf{c}_2) = 2 < 8.$$

Thus, the Hamming-metric tracing algorithm can find the traitor $\mathbf{c}_2$. But it fails to find the traitor $\mathbf{c}_1$.

Now, for $u = 1$ by Theorem 2, the Lee-metric tracing algorithm has a capability

$$\tau = (u+1)(n - \left\lfloor \sqrt{n(k-1)} \right\rfloor - 1) = 10.$$

Thus, the Lee-metric tracing algorithm can find all the traitors satisfying

$$d_L(\mathbf{x}, \mathbf{c}) \le 10.$$

It is easy to verify that

$$d_L(\mathbf{x}, \mathbf{c}_1) = 10, \quad \text{and} \quad d_L(\mathbf{x}, \mathbf{c}_2) = 2.$$

Therefore, the Lee-metric tracing algorithm finds both the traitors $\mathbf{c}_1$ and $\mathbf{c}_2$.

5.2 Performance and Complexity

According to [16] the traitor-tracing performance of the Hamming-metric tracing algorithm is

$$\tau_H = \begin{cases} n - \left\lfloor \sqrt{n(k-1)} \right\rfloor - 1 \\ \qquad \text{for RS-traceability codes,} \\ n - \left\lfloor \sqrt{n(k+g-1)} \right\rfloor - 1 \\ \qquad \text{for AG-traceability codes.} \end{cases} \quad (5.1)$$

While, by Theorem 2 the performance of the Lee-metric tracing algorithm is

$$\tau_L = \begin{cases} (u+1)(n - \left\lfloor \sqrt{(2u+1)n(k-1)} \right\rfloor - 1) \\ \qquad \text{for RS-traceability codes,} \\ (u+1)(n - \left\lfloor \sqrt{(2u+1)n(k+g-1)} \right\rfloor - 1) \\ \qquad \text{for AG-traceability codes.} \end{cases} \quad (5.2)$$

Here u can be any integer with $0 \le u \le (p-1)/2 - 1$, where p is the size of the alphabet. The fact that u is variable in the interval $[0, \ (p-1)/2 - 1]$ gives us more flexibility to maximize the traitor-tracing performance. This is the reason why the Lee-metric tracing algorithm can find more traitors than the Hamming-metric tracing algorithm.

In the following we will compare the algorithms by plotting τ_H and τ_L (which will actually be normalized against the code length) in a single figure (Fig. 1). It is well known that (see [16] and [17] for example) only those RS and AG codes which have very low code rate can be used as traceability codes. So, in Fig. 1 we only plot normalized τ_H and τ_L for small code rates. More precisely, for a RS or AG code to be used as a t-traceability code, a sufficient condition is $d > n - n/t^2$ ([16], [17]). As the code dimension $k \le n - d + 1$, we have $\delta < (1/t^2) + (1/n)$, where $\delta = k/n$ is the code rate. In most practical applications, the preferable n is usually in the range between 100 and a few thousands; and t is greater than or equal to 3. The following table (Table 1) shows the code rates δ for some values of n and t.

In Fig. 1 the vertical axis represents normalized traitor-tracing performance against the code length, while horizontal axis represents code rate δ. From the figure we can see that when the code has a low rate (say, $\delta \le 0.165$), the Lee-metric tracing algorithm has a better performance than the traditional tracing algorithm.

Table 1: Rates of RS-traceability codes

	$n = 100$	$n = 200$
$n = 3$	$\delta < 0.1212$	$\delta < 0.1178$
$n = 4$	$\delta < 0.0725$	$\delta < 0.0692$
$n = 5$	$\delta < 0.05$	$\delta < 0.0467$
$n = 6$	$\delta < 0.0378$	$\delta < 0.0328$

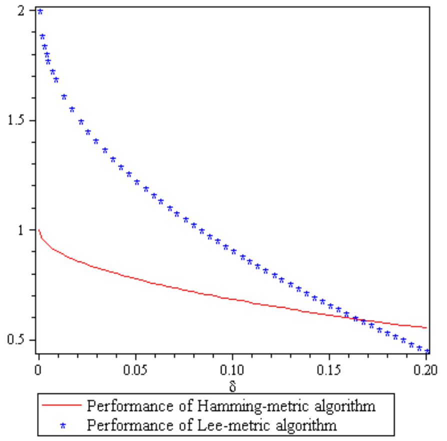

Fig. 1: Performance comparison of tracing algorithms

Now we consider the computational complexity. Note that for the Lee-metric tracing algorithm, Algorithm 1, the dominant steps in complexity are the interpolation step (i.e., Step 2) and the factorization step (i.e., Step 3). It is well known that (see [16], [20]) the complexity of finding a nonzero bivariate polynomial by an interpolation procedure over 2n(u+1) points is O(n3u3) = O(n3) (as u is usually a small constant). Making use of an improved procedure in [13], the complexity of Step 2 can be reduced to O(n2). Using an efficient root-finding procedure in [13], [20], the codeword polynomials can be found with Step 3 in complexity O(c(log c)2kn) which is bounded from above by O(c2n2), where c is the size of the output list of the algorithm. It is easy to see that c=O(t) where t is the size of the coalition. Thus, the overall complexity of the Lee-metric tracing algorithm is O(t2n2). This is the same as the complexity of the Hamming-metric tracing algorithm in [16].

6. Conclusion

By generalizing traditional traceability codes, we have derived a new class of traceability codes. The generalized traceability codes still retain the identifiable parent property; and they contain new traceability codes. By adapting a decoding algorithm with respect to the Lee distance that we proposed in our previous work, we have presented an efficient tracing algorithm for the new traceability codes. The proposed algorithm can outperform the Hamming-metric tracing algorithm, and has the same complexity as the Hamming-metric tracing algorithm.

References

[1] A. Barg, G. Blakley, and G. Kabatiansky, “Digital fingerprinting codes: problem statements, constructions, identification of traitors,” IEEE Trans. Inform. Theory, 49, (4), pp. 852-865, 2003.

[2] D. Boneh, and J. Shaw, “Collusion secure fingerprinting for digital data,” IEEE Trans. Inform. Theory, 44, (5), pp. 1897-1905, 1998.

[3] E. Candes, and T. Tao, “Decoding by linear programming,” IEEE Trans. Inform. Theory, 2005, 51, (12), pp. 4203-4215.

[4] B. Chor, A. Fiat and M. Naor, “Tracing traitors,” Advances in Cryptology - Crypto'94 (Lecture Notes in Computer Science), Berlin, Germany: Springer-Verlag, vol.839, pp.480-491, 1994.

[5] B. Chor, A. Fiat, M. Naor, and B. Pinkas, “Tracing traitors,” IEEE Trans. Inform. Theory, 46, (3), pp. 893-910, 2000.

[6] M. Fernandez, and M. Soriano, “Algorithm to decode `identifiable parent property' codes,” Electronics Letters, 38, (12), pp. 552-553, 2002.

[7] M. Fernandez, M. Soriano, and J. Cotrina, “Tracing illegal redistribution using error-and-erasures and side information decoding algorithms,” IET Inf. Secur., 2, (1), pp.83-90, 2007.

[8] A. Fiat, and T. Tassa, “Dynamic traitor tracing,” J. Cryptology, 14, pp.211-223, 2001.

[9] V. Guruswami, and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometry codes," IEEE Trans. Inform. Theory, 45, (6), pp. 1757-1767, 1999.

[10] H. Hollmann, J.H. van Lint, J.-P. Linnartz, and L. Tolhuizen, "On codes with the identifiable parent property," J. Comb. Theory A, vol.82, pp. 121-133, 1998.

[11] C. Kamath, Scientific Data Mining - A Practical Perspective, SIAM, Philadelphia, USA, 2009.

[12] J. Mee, and P. Watters, "Detecting and tracing copyright infringements in P2P networks," Proceedings of Fifth International Conference on Networking and the International Conference on Systems, 23-29 April 2006, Mauritius, pp. 60-65.

[13] R. Roth, and G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," IEEE Trans. Inform. Theory, 46, (1), pp. 246-257, 2000.

[14] R. Roth, and P. Siegel, "Lee-Metric BCH Codes and their Application to Constrained and Partial-Response Channels," IEEE Trans. Inform. Theory, 40, (4), pp.1083-1096, 1994.

[15] R. Safavi-Naini, and Y. Wang, "Sequential traitor tracing," IEEE Trans. Inform. Theory, 49, (5), pp.1319-1326, 2003.

[16] A. Silverberg, J. Staddon, and J. Walker, "Applications of list decoding to tracing traitors," IEEE Trans. Inform. Theory, 49, (5), pp.1312-1318, 2003.

[17] J. Staddon, D. Stinson, and R. Wei, "Combinatorial properties of frameproof and traceability code," IEEE Trans. Inform. Theory, 47, (3), pp.1042-1049, 2001.

[18] X.-W. Wu, M. Kuijper, and P. Udaya, "Lee-Metric Decoding of BCH and Reed-Solomon Codes," Electronics Letters, 39, (21), pp.1522-1524, 2003.

[19] X.-W. Wu, M. Kuijper, and P. Udaya, "On the Decoding Radius of Lee-Metric Decoding of Algebraic-Geometric Codes," Proceedings of 2005 IEEE International Symposium on Information Theory, Adelaide, Australia, 5-9 September, 2005, pp.1191-1195.

[20] X.-W. Wu, and P. Siegel, "Efficient Root-Finding Algorithm with Applications to List Decoding of Algebraic-Geometric Codes," IEEE Trans. Inform. Theory, 47, (6), pp. 2579-2587, 2001.

[21] X.-W. Wu, P. Watters, and J. Yearwood, "New Traceability Codes and Identification Algorithm for Tracing Pirates," Proceedings of 2008 IEEE International Symposium on Parallel and Distributed Processing with Application, Sydney, Australia, 10 - 12 December, 2008, pp.719-724.

Xin-Wen Wu received the Ph.D. degree from the Chinese Academy of Sciences. He was with the Institute of Mathematics, Chinese Academy of Sciences, the University of Louisiana (as a visiting researcher), the University of California, San Diego (as a post-doctoral researcher), and the University of Melbourne (holding a research fellowship). From November 2005 to April 2010, he was an academic staff member at University of Ballarat; he then has been a faculty member of the School of Information and Communication Technology, Griffith University, Australia. His research interests include coding and information theory, applied cryptography, and the applications. He has published extensively in these areas. He is a member of IEEE.

Abdul Sattar is the founding Director of the Institute for Integrated and Intelligent Systems and a Professor of Computer Science and Artificial Intelligence at Griffith University. He has been at Griffith University since February 1992, and appointed as a professor in 2000, within the School of Information and Communication Technology. He is also a research leader at NICTA Queensland Research Laboratory. His research interests include knowledge representation and reasoning, constraint satisfaction, intelligent scheduling, rational agents, propositional satisfiability, temporal reasoning, temporal databases, data security, and bioinformatics. Prof. Sattar is a life member of the Association for the Advancement of Artificial Intelligence (AAAI), and a member of several professional bodies including the International Society for Applied Intelligence, the Association of Computing Machinery (ACM) and the International Artificial Intelligence in Education Society.